

```
#include<iostream>
#include<string.h>
#include<stdlib.h>

using namespace std;

typedef struct {
    char nome[30];
    int idade;
    int ra;
    float nota;
} TipoItem;

typedef struct No *Apontador;

typedef struct No {
    TipoItem item;
    Apontador prox;
} NoAluno;

typedef struct {
    Apontador Primeiro, Ultimo;
} TipoLista;

void criaLista(TipoLista *Lista);
void insereNo(TipoItem aluno, TipoLista *Lista);
void removeNo(int ra, TipoLista *Lista);
NoAluno* buscaNo(int ra, TipoLista *Lista);
void imprimeLista(TipoLista *Lista);

main(){
    TipoItem aluno;
    TipoLista Lista;

    // cria lista dinâmica
    criaLista(&Lista);

    // inserção
    cout << endl << "INSERINDO RA: 1234" << endl;
    strcpy(aluno.nome, "Teste");
    aluno.idade=23;
    aluno.ra=1234;
    aluno.nota = 8.5;
    insereNo(aluno,&Lista);

    // inserção
    cout << "INSERINDO RA: 2345" << endl;
    strcpy(aluno.nome, "Teste2");
    aluno.idade=25;
    aluno.ra=2345;
    aluno.nota = 6.5;
    insereNo(aluno,&Lista);

    // inserção
    cout << "INSERINDO RA: 5678" << endl;
    strcpy(aluno.nome, "Teste3");
    aluno.idade=27;
    aluno.ra=5678;
    aluno.nota = 4.5;
    insereNo(aluno,&Lista);

    // imprime lista
    imprimeLista(&Lista);

    // testa busca
    cout << endl << "PROCURANDO RA: 2345" << endl;
    NoAluno *p = buscaNo(2345,&Lista);
    cout << "ENCONTRADO! Nome: " << p->item.nome << " Idade: " << p->item.idade << " RA: " << p->item.ra << " Nota: " << p->item.nota << endl;

    // remoção
    cout << endl << "REMOVENDO RA: 2345" << endl;
```

```
removeNo(2345,&Lista);

// imprime lista
imprimeLista(&Lista);
cout << endl;
}

void criaLista(TipoLista *Lista){
    Lista->Primeiro = (Apontador) malloc(sizeof(NoAluno)); // aloca espaço para nó cabeça
    Lista->Ultimo = Lista->Primeiro; // equivalente a (*Lista).Ultimo = (*Lista).Primeiro;
    Lista->Primeiro->prox = NULL; // é obrigatório atribuir NULL
}

void insereNo(TipoItem aluno, TipoLista *Lista){
    Lista->Ultimo->prox = (Apontador) malloc(sizeof(NoAluno)); // aloca espaço para novo nó
    Lista->Ultimo = Lista->Ultimo->prox;
    Lista->Ultimo->item = aluno;
    Lista->Ultimo->prox = NULL;
}

void imprimeLista(TipoLista *Lista){
    Apontador p;
    p = Lista->Primeiro->prox;
    cout << endl << "IMPRESSÃO DA LISTA" << endl;
    while (p!=NULL){
        cout << "Nome: " << p->item.nome << " Idade: " << p->item.idade << " RA: " << p->item.ra << "
Nota: " << p->item.nota << endl;
        p = p->prox;
    }
}

void removeNo(int ra, TipoLista *Lista){
    Apontador p,q;
    p = Lista->Primeiro->prox;
    q = Lista->Primeiro->prox;
    while ((p!=NULL) && (p->item.ra!=ra)){
        q = p;
        p = p->prox;
    }
    if (p!=NULL){
        q->prox = p->prox;
        free(p);
    } else {
        cout << "Registro não encontrado para remoção!" << endl;
    }
}

NoAluno* buscaNo(int ra, TipoLista *Lista){
    Apontador p;
    p = Lista->Primeiro->prox;
    while ((p!=NULL) && (p->item.ra!=ra)){
        p = p->prox;
    }
    if (p!=NULL){
        return p;
        //cout << "ENCONTRADO! Nome: " << p->item.nome << " Idade: " << p->item.idade << " RA: " << p->item.ra << " Nota: " << p->item.nota << endl;
    } else {
        return NULL;
        //cout << "Registro não encontrado!" << endl;
    }
}
```